PATENT

UNITED STATES PATENT APPLICATION

for

LOAD BALANCING MODEL FOR MULTILINK FRAME RELAY

Inventors:

Chang Ahn
Swaminathan Sundararaman

prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Los Angeles, CA 90026-1026
(408) 720-8300

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number: ______________________________
Date of Deposit: ____
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to BOX PATENT APPLICATION, Assistant Commissioner for Patents, Washington, D. C. 20231

____
(Typed or printed name of person mailing paper or fee)

____
(Signature of person mailing paper or fee)

____
(Date signed)

LOAD BALANCING MODEL FOR MULTILINK FRAME RELAY

FIELD OF THE INVENTION

[0001] The field of the invention relates to telecommunication. More specifically, it relates to load balancing for multilink frame relay connections.

BACKGROUND OF THE INVENTION

[0002] Multilink frame relay connections provide several advantages over the traditional asynchronous transfer mode (ATM) of communication. While ATM can be used to send data over multiple links, by using inverse multiplexing over ATM (IMA), usually the data cells sent must all be of the same size. Additionally, ATM usually requires the links to have the same speed of transfer. Multilink has no such limitations to size and speed.

[0003] Large frames are usually fragmented to prevent clogging of the bundle link. A timer is activated when fragments arrive out of sequence, with interior fragments missing. If the timer finishes before a missing fragment reaches the destination, the entire frame is lost.

[0004] Previously, multilink connections usually used either a round robin or a credit method to determine assign data frames or data frame fragments. Exemplary results of both the round robin method and the credit method are illustrated in Figures 1a and 1b. In this example, a transmitter 100 breaks the data frame into six fragments, labeled 101, 102, 103, 104, 105, and 106. The multilink data frame relay connection has three bundle links, each of separate speed ratings. Link 110 has a rating of 1ds0, link 120 is twice as fast with a rating of 2ds0, and link 130 is three times as fast with a rating of

3ds0. A receiver 140 receives the frame fragments and reassembles the frame. In one embodiment, the transmitter and the receiver are transceivers.

[0005] In the round robin method, illustrated in Figure 1a, a transmitter 100 transmits data frame fragments chronologically into each bundle link, repeating when all links have been used. This method results in fragments 101 and 104 being transmitted in link 110, fragments 102 and 105 being transmitted in link 120, and fragments 103 and 106 being transmitted in link 130. Because of the differentiated speed between links, the order of arrival is then 103, 102, 106, 101, 104, and 105, with 101, 104, and 105 all arriving simultaneously. The fragments arriving in this order would activate the timer, and possibly the whole frame would be lost.

[0006] In the credit method, illustrated in Figure 1b, bundle links are weighted according to link speed. Frames and frame fragments are sent on the fastest link, link 130 in this example, until a threshold is exceeded, at which point the frames and frame fragments are sent on the next fastest link, link 120. This method results in frame fragments 101, 102, and 103 being transmitted on link 130; frame fragments 104 and 105 being transmitted on link 120; and frame fragment 106 being transmitted on link 110. The order of arrival becomes 101, 104, 102, 106, 105, and 103, with 106, 105, and 103 arriving simultaneously. This order also activates the timer and endangers capture of the entire frame.

[0007] The reason for this is that, while link 130 transfers data much faster than either link 120 or link 110, as more frame and frame fragments are transmitted through a link, the link is slowed down. The phenomenon is similar to lanes in a freeway. The carpool

lane may allow a car to travel faster, but as more cars load into the lane, the speed slows down.

SUMMARY OF THE INVENTION

[0008] A system is described that includes a transmitter to send data frames over a multilink data connection. A receiver receives data frames over the multilink data connection. The multilink data connection contains a set of individual links given a credit value based on each link's speed of data transmission and current level of data traffic.

[0009] Other features and advantages of the present invention will be apparent from the accompanying drawings and from the detailed description that follows below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicated similar elements and in which:

[0011] Figures 1a and 1b illustrate prior methods of load distribution for a multilink frame relay connection.

[0012] Figure 2 illustrates variations in band throughput for different bundle links in a multilink connection.

[0013] Figure 3 illustrates the final distribution of fragments when using the system.

[0014] Figure 4 is a flowchart of a method for setting up a system to send a balance load through a multilink frame relay connection.

[0015] Figure 5 is a flowchart of a method for sending a frame through the multilink frame relay connection.

[0016] Figure 6 is a flowchart of a method for choosing a link to send a fragment of a frame of data through in a multilink frame relay connection.

[0017] Figure 7 is a flowchart of a method for resetting the system.

DETAILED DESCRIPTION

[0018] A system and method are described for load balancing the transmission of data frames across a multilink connection. Links in a multilink connection are rated for speed. A credit value is assigned to each bundle link based on the amount of data that can be transmitted across a line in a set period of time. When a frame is sent across a multilink connection, the first frame or frame fragment is sent across the bundle link with the highest credit value. If two links have the same credit value, the frame is sent across the slower of the two links. Once a frame or frame fragment is sent, the credit value of that link is reduced by the size of the frame or frame fragment. All the credit values are reset when one link has a negative credit value or all links have a credit value equal to zero. If the multilink connection is unused for a set period of time, all the credit values are reset.

[0019] For one embodiment, a credit value is assigned to each link based on how much data can be transmitted through the link in a given period of time. An example of the differing rates at which data can be transferred and how this difference affects credit values is illustrated in Figure 2. For one embodiment, a time period 200 is chosen in relation to the time required to transfer a frame 211 of a given length 210 over the slowest link 110. For one embodiment, if the length 210 is 80 bytes, and only one frame 211 is able to pass over link 110, link 110 will have a credit value of 80. Link 120, which is twice as fast as link 110, can transmit equivalent frames 212 and 213 in the same period of time 200. Therefore, link 120 has a rating of 160. Link 130, which is three times as fast as link 110, can transmit equivalent frames 214, 215, and 216 in the same period of time 200. Therefore, link 130 has a credit value of 240.

[0020] The results of these credit values produce an exemplary distribution illustrated in Figure 3. As link 130 has the highest credit value with 240, the first frame fragment 101 is transmitted on that link. If the frame fragment has a size of 80 bytes, then the credit value is reduced by that amount. Now link 130 and link 120 both have a credit value of 160. Frame fragment 102 is transmitted on link 120, as link 120 is slower than 130. As link 130 again has the highest credit value with 160, frame fragment 103 is transmitted on the link. As all links now have a credit value of 80, the frames are sent on the slowest available link, with frame fragment 104 on link 110, frame fragment 105 on link 120, and frame fragment 106 on link 130. The receiver 140 receives the frame fragments in the order 101, 102, 103, 104, 105, and 106.

[0021] One embodiment of a method is illustrated in Figure 4. A time period (T) is assigned over which to measure data throughput and to initiate the timed reset 400. For other embodiments, separate time periods are chosen to measure throughput and to initiate resets. The clock (t) is reset to zero 410. A link index (L), by which to count links from 0 to *n*, with *n*+1 representing the total number of links, is set to 0 to initialize the links 420. The speed of data transmission (v) is determined for link 0 430. The current credit (CC) for link 0, equal to speed times the time period, is determined and assigned as the initial credit value (C0) 440. The credit taken (CCT) is set to 0 450. As long as not all the links have been initialized 460, the index is incremented 470, and the initialization begins again for the next link at the first initialization step 430. After initialization is complete, the process is in sleep mode until a frame is available to be sent 480. Once a frame is available, the frame is transmitted 500. If the time period

has not expired once the transmission is completed 490, the process goes back into sleep mode 480. Otherwise, the credits on the links are reset 700.

[0022] One embodiment of a method of transmitting the frames is illustrated in Figure 5. To send the frame 500, the frame is first divided into $k+1$ number of frame fragments, where $k+1$ is a preset number 510. For other embodiments, $k+1$ is dynamically determined based on the size of the frame. A fragment index (M) is initialized to zero to count the fragments from 0 to k 520. A link choosing subroutine returns the index of the link on which the fragment will be transmitted 600. The fragment is transmitted on the chosen link 530. Current credit (CC) is reduced by the size (FS) of the fragment 540. The current credit taken (CCT) is increased by the fragment size (FS) 550. If the current credit is less than zero 560, all the link credits are reset 700. If current credit is not less than zero and more fragments need to be transmitted 570, a new link is chosen 600. If no fragments are left to be transmitted 570, the process is ready to send the next frame 580.

[0023] One embodiment of a method of choosing a link is illustrated in Figure 6. The choose link process is initiated each time a frame fragment is to be sent 600. The current credit of link 0 is tested to see if the value is at zero 605. If the current credit is at zero, the flag is set equal to 0 610. If the current credit is not zero, the flag is set equal to 1 615. An index (NL) to indicate the chosen link to return is set to indicate link 0 620. A regular link index (L) is initialized to 1 to increment through links 1 to n, where $n+1$ represents the number of links 625. The current credit of link NL is compared with the current credit of link L 630. If the current credit of link L is greater than the current credit of link NL, the flag is set to 1 635 and the chosen link index NL is set equal to L

640. If the current credit of NL is greater than the current credit of link L, the chosen link index remains NL. If the current credits of links L and NL equal each other, the speed (v) of link L is then compared with the speed (v) of link NL 645. If link NL is faster, the chosen link index is set to link L 640. If link L is faster, the index remains the same. If more links exist 650, the index L is incremented 655 and the new link L is compared to link NL 630. After all the links have been compared, the flag is checked to see if the value is still zero 660. If the value is still zero, the indication is that all of the links have a current credit of zero. The credits are reset 700, the flag is set to one 665, and the chosen link index is reset to zero 620. The comparison is then repeated with the reset current credits. After the link is chosen, the chosen link's index is returned to the main process 670.

[0024] One embodiment of a reset method is illustrated in Figure 7. An outside process implements the reset method, either because the time period (T) has expired, one of the links has a negative current credit, or all the links had zero current credit 700. A link index (L) is initialized to 0 to increment links 0 through n, where $n+1$ represents the number of links 710. The current credit (CC) of the link is set to the initial credit (C0) determined when the system was initialized 720. In an alternate embodiment, the current credit (CC) is set equal to the sum of that link's current credit plus the current credit taken (CCT) 720. The current credit taken (CCT) is set to zero 730. If a link has not been reset 740, the link index is incremented 750 and the next link is reset 720. After all the links have been reset 740, the clock is reset to zero 760. The system is then ready for the next frame 770.

[0025] The method and apparatus disclosed herein may be integrated into advanced Internet-based or network-based knowledge systems as related to information retrieval, information extraction, and question and answer systems.

[0026] The method described above can be stored in the memory of a computer system (e.g., set top box, video recorders, etc.) as a set of instructions to be executed. The instructions to perform the method described above could alternatively be stored on other forms of machine-readable media, including magnetic and optical disks. For example, the method of the present invention could be stored on machine-readable media, such as magnetic disks or optical disks, which are accessible via a disk drive (or computer-readable medium drive). Further, the instructions can be downloaded into a computing device over a data network in a form of compiled and linked version.

[0027] Alternatively, the logic to perform the methods as discussed above, could be implemented by additional computer and/or machine readable media, such as discrete hardware components as large-scale integrated circuits (LSI's), application-specific integrated circuits (ASIC's), firmware such as electrically erasable programmable read-only memory (EEPROM's); and electrical, optical, acoustical and other forms of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

[0028] Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.